

ALZA, J., BARTLETT, M., CEBERIO, J. and MCCALL, J. 2021. Towards the landscape rotation as a perturbation strategy on the quadratic assignment problem. In *Chicano, F. (ed.) GECCO '21: proceedings of 2021 Genetic and evolutionary computation conference companion*, 10-14 July 2021, [virtual conference]. New York: ACM [online], pages 1405-1413. Available from: <https://doi.org/10.1145/3449726.3463139>

Towards the landscape rotation as a perturbation strategy on the quadratic assignment problem.

ALZA, J., BARTLETT, M., CEBERIO, J. and MCCALL, J.

2021

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Towards the Landscape Rotation as a Perturbation Strategy on the Quadratic Assignment Problem

Joan Alza
Robert Gordon University
Aberdeen, Scotland
j.alza-santos@rgu.ac.uk

Josu Ceberio
University of the Basque Country (UPV/EHU)
Donostia, Spain
josu.ceberio@ehu.eus

Mark Bartlett
Robert Gordon University
Aberdeen, Scotland
m.bartlett3@rgu.ac.uk

John McCall
National Subsea Centre
Aberdeen, Scotland
j.mccall@rgu.ac.uk

ABSTRACT

Recent work in combinatorial optimisation have demonstrated that neighbouring solutions of a local optima may belong to more favourable attraction basins. In this sense, the perturbation strategy plays a critical role on local search based algorithms to *kick* the search of the algorithm into more prominent areas of the space. In this paper, we investigate the landscape rotation as a perturbation strategy to redirect the search of an stuck algorithm. This technique rearranges the mapping of solutions to different objective values without altering important properties of the problem's landscape such as the number and quality of optima, among others. Particularly, we investigate two rotation based perturbation strategies: (i) a profoundness rotation method and (ii) a broadness rotation method. These methods are applied into the stochastic hill-climbing heuristic and tested and compared on different instances of the quadratic assignment problem against other algorithm versions. Performed experiments reveal that the landscape rotation is an efficient perturbation strategy to shift the search in a controlled way. Nevertheless, an empirical investigation of the landscape rotation demonstrates that it needs to be cautiously manipulated in the permutation space since a small rotation does not necessarily mean a small disturbance in the fitness landscape.

CCS CONCEPTS

• **Mathematics of computing** → *Permutations and combinations; Evolutionary algorithms*; • **Theory of computation** → *Evolutionary algorithms*.

KEYWORDS

Quadratic Assignment Problem, Landscape Rotation

ACM Reference Format:

Joan Alza, Mark Bartlett, Josu Ceberio, and John McCall. 2021. Towards the Landscape Rotation as a Perturbation Strategy on the Quadratic Assignment Problem.

1 INTRODUCTION

In evolutionary computation, algorithms are commonly tested and evaluated on a set of problem instances with different properties. In order to intuit the trajectory or suitability of the algorithms on the optimised instance, researchers employ the metaphor called *fitness landscape*. Informally speaking, a fitness landscape is the representation of an abstract space generated by the objective value assigned to each solution and a specific set of variation operations that connect them (usually represented by neighbourhood structures).

Many authors have found it practical to study features of the fitness landscape since they seem to condition the behaviour of the algorithms [14, 18, 20]. Abstractly, these characteristics affect the geometric properties of the landscape such as the arrangement of solutions among attraction basins. In brief, a basin of attraction of an optima is a subset of connected solutions that lead to a local optima when a steepest ascent hill-climbing algorithm is applied. Hernando et al. in [6] study the shape of the fitness landscape and the attraction basins' anatomy on combinatorial problems, and suggest how to enhance the design of future algorithms. The authors mention that local search based algorithms get stuck on plateaus or a local optima, although these are generally connected to neighbouring solutions that belong to more favourable basins of attractions. Therefore, even one movement from a local optima is enough to explore other areas (in terms of attraction basins) of the space.

Literature presents a wide variety of perturbation strategies to avoid getting trapped in a given attraction basin [6]. In this paper, we use the fitness landscape rotation as a perturbation technique to redirect an algorithm that is stuck in a local optima. Initially presented as a dynamic benchmark generator for the binary space [23, 24], this method also stands in the permutation space [16, 25], or even in the continuous domain [11]. It generates a sequence of static problem instances by incrementally inserting modifications to an initial static problem instance. A rotation of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '21 Companion, July 10–14, 2021, Lille, France

© 2021 Association for Computing Machinery.

fitness landscape consist of relabelling the elements that form the solutions in the search space. Such relabelling provokes that each solution is mapped to the objective value of another solution of the original problem. The popularity of this method comes from its simplicity, but especially from its capability to preserve important properties of the fitness landscape (or problem instance) such as the number and quality of the optima or the neighbourhood relations among the solutions. However, as far as we know, there is a limited knowledge about the hidden insights of the landscape rotation in the permutation space.

This work briefly investigates the landscape rotation in the permutation domain theoretically, and presents the landscape rotation as a perturbation technique to *kick* the search of a stuck algorithm into other attraction basins. In particular, two rotation based perturbation strategies are suggested: (i) a profoundness rotation strategy is presented, where the already found optima is compared with the obtained optima on rotated environments; (ii) a broadness rotation strategy that spends some time on the rotated space before returning to the original environment is designed.

In order to evaluate the performance of the proposed strategies, two types of experiments are conducted using the quadratic assignment problem (QAP) [10] as case study. First, the rotation parameters of the algorithms are tuned up, i.e. the time to spend exploring the neighbourhood and the rotation degree need to be adjusted beforehand. Second, we compare their performance against the classical stochastic hill-climbing heuristic (sHC) and its multi-starting version (sHC-r). Obtained results demonstrate the effectiveness of rotating the landscape to explore prominent areas of the space. Moreover, results reveal that the symmetries of the QAP problem instances clearly influence the efficiency of the proposed algorithms: while in symmetric instances the proposal is preferred, sHC-r is, in general, better for asymmetric instances.

The rest of the paper is organised as follows. The QAP and the background of the permutation space are formally presented in Section 2. Then, Section 3 introduces and analyses the landscape rotation on the permutation domain. Section 4 reveals the designed landscape rotation based algorithms in detail. The experiments and results are described and analysed in Section 5, and the paper concludes in Section 6.

2 PERMUTATION PROBLEMS

A permutation is a bijection from a set onto itself, and they are commonly used to represent the ordering of n items. The set of size n , commonly known as the *symmetric group* \mathbb{S}_n , is composed by $n!$ permutations. Permutations are usually denoted as $\sigma, \pi \in \mathbb{S}_n$, where $\sigma(i)$ stands for the item at position i in permutation σ . A special property of permutations is the identity permutation e , where $1 \leq i \leq n, e(i) = i$. The composition between two permutations returns a new one by $1 \leq i \leq n, \pi \circ \sigma = \pi(\sigma(i))$. In general, the composition between permutations is not commutative, so $\sigma \circ \pi \neq \pi \circ \sigma$.

The optimisation problems that are encoded by permutations of n items are often named *permutation problems* in the literature. Typical permutation problems are the travelling salesman problem (TSP) [8], the linear ordering problem (LOP) [4, 15] or the quadratic assignment problem (QAP) [10].

Koopmans and Beckman introduced the QAP as an unconstrained permutation problem that consists of assigning a set of facilities to a set of locations such that the total assignment cost is minimised. Formally, the problem consists of a distance matrix D and a flow matrix F , both of size $n \times n$, where $d_{x,y} \in D$ is the distance between locations x and y , and $f_{i,j} \in F$ is the flow between facilities i and j . The total assignment cost, represented by a permutation σ , is calculated as:

$$f(\sigma) = \sum_{i=1}^n \sum_{j=1}^n f_{i,j} d_{\sigma(i), \sigma(j)}. \quad (1)$$

Many QAP instances used in academia contain symmetries in their solution space due to the symmetrical patterns of flow and distance matrices. As studied in [5], such symmetries exist when the locations of the facilities are grouped in a rectangular way.

3 LANDSCAPE ROTATION

In dynamic optimisation, benchmark generators are commonly used to create problem sequences to compare and evaluate the performance of algorithms in a dynamic way. One of the most popular dynamic benchmark generator in the combinatorial space is the *fitness landscape rotation*. This method goes back to [23, 24], where the XOR DOP benchmark generator was introduced as a simple and straightforward way to periodically change a static binary problem. Formally, given a rotation frequency τ (number of iterations between changes) and a rotation degree ρ , the objective value of a solution x is periodically changed through the exclusive-or operation between the solution and a binary mask m . Therefore, the solution x is mapped to the value of the solution $x \oplus m$ when the problem is changed. Analysed by Tinós and Yang [21, 22], the XOR DOP generator permutes the problem in a special way, so important problem properties remain the same over time, i.e. the number and quality of optima or the neighbourhood relations between solutions are preserved.

In [25], Younes et al. extended the landscape rotation to the permutation space, and presented the generalised benchmark generator (GBG). This technique encodes the sequence of static instances as

$$S = \{(I_j, x_j), j \in (1, k)\}, \quad (2)$$

where k is the size of the sequence, I_j is the problem instance at change period j and x_j is a collection of the optimal solutions at change period j . The instances are incrementally generated by

$$I_j = I_{j-1} \oplus \Delta_j, \quad (3)$$

where Δ_j is an environmental shift variable that is applied to the instance. Note that I_1 represents the initial problem instance. The rotation degree $\rho \in (0.0, 1.0]$ represents the number of elementary steps added to create Δ_j , where $\lceil n \times \rho \rceil$ is the number of exchanges imposed on the mapping function.

Nevertheless, Mavrovouniotis et al. [16] comment that rotating the permutation landscape at $\rho = 0.5$ using GBG might reorder all the elements (swap half of the variables with the other half) and, thus, provoke an unintended change. In any case, as pointed out by the authors in [1], the swaps between variables in the permutation domain can be measured as the Cayley distance. Moreover, other permutation metrics can be used to generate other types of modifications apart from swaps (we guide the interested reader to [7]

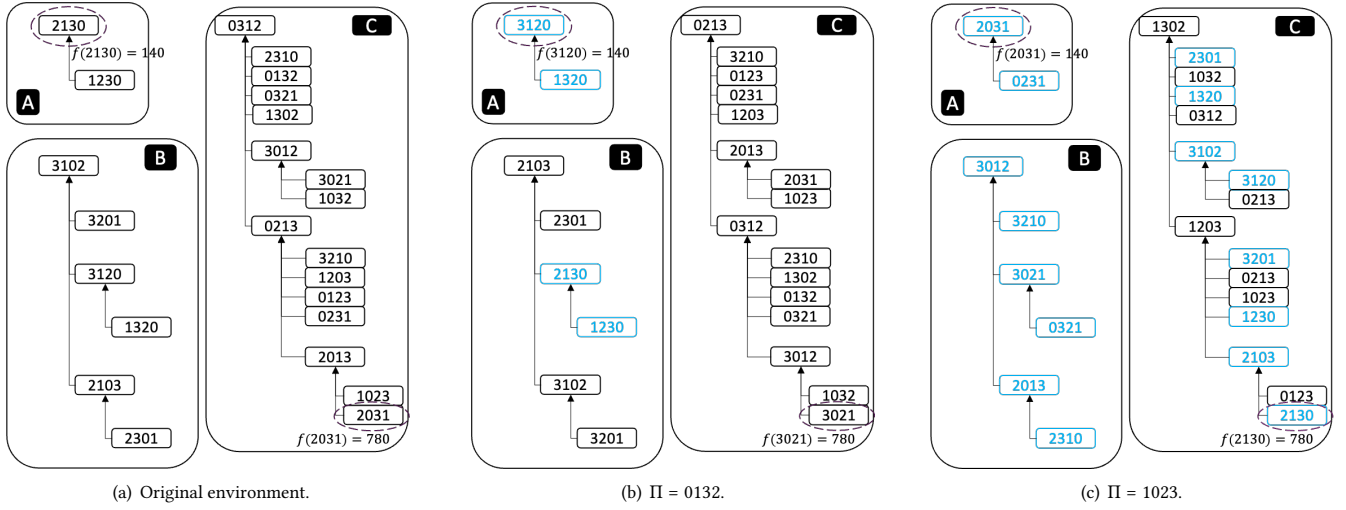


Figure 1: Attraction basins of a QAP instance of size $n = 4$ under the 2-exchange neighbourhood. The coloured cells represent the solutions that migrate between the attraction basins compared to the original environment.

for more information about permutation distance metrics). In any case, the number of steps to change the instance varies with the maximum and minimum distances of the metric considered, so that

$$d = \begin{cases} \lceil d_{\max}^K \times \rho \rceil \in \{1, \dots, \binom{n}{2}\}, & \text{Kendall's-}\tau \ (d^K) \\ \lceil d_{\max}^H \times \rho \rceil \in \{2, \dots, n\}, & \text{Hamming} \ (d^H) \\ \lceil d_{\max}^C \times \rho \rceil \in \{1, \dots, n-1\}, & \text{Cayley} \ (d^C) \text{ or Ulam} \ (d^U), \end{cases}$$

where d_{\max} is the maximum distance of a given metric and n is the size of the permutations.

To the best of our knowledge, there are no works that studied in detail the landscape rotation in the permutation space, in the same manner as Tinós and Yang did in the binary space [21, 22]. For that reason, the following section collects some insights about the landscape rotation in the permutation domain by studying the topology of the fitness landscape before and after rotating the landscape.

3.1 Findings

Formally, the objective function on a rotated permutation space, f' , can be seen as follows:

$$f'(\sigma) = f_s(\Pi \circ \sigma), \quad (4)$$

where $\sigma \in \mathbb{S}_n$ is a permutation, f_s is the stationary objective function and Π is a permutation mask containing $\lceil m \times \rho \rceil$ steps from the identity permutation e (similar to Δ_j in Equation 3). The permutation Π is generated uniformly at random at a given distance, under a metric, following the theoretical basis presented in [7]. In short, given a permutation size n and a distance $d = \lceil m \times \rho \rceil$, a uniformly at random permutation is generated at distance d from the identity permutation e . In this way, the solution σ will be mapped to the value of $\Pi \circ \sigma$ in the rotated environment.

3.1.1 Structure of the Space. This assumption does not alter the fitness landscape structure, so the ranking of the solutions in the search space (in terms of their objective values) remains the same

even when the landscape is rotated. Let's illustrate it with an example of a QAP instance of size $n = 4$. Figure 1(a) shows the structure of the fitness landscape as a directed graph that represents the attraction basins under the 2-exchange neighbourhood. The nodes of the graph represent the solutions (permutations), and the directed edges represent the movement to the best neighbour when a steepest ascent hill-climbing algorithm is applied. Figure 1(b) & 1(c) show the attraction basins after rotating the original landscape by some degree. As can be observed, the solutions are relocated through the space, although the structure of the space, the number and arrangement of the attractions basins and the number and quality of optima are preserved. Thus, we can deduce that solutions are mapped to different objective values, although the arrangement of solutions within the attraction basins, in terms of objective values, remain the same. Continuing with the previous example, the solution at the lowest attraction basins level in C is always mapped to the objective value 780, and similarly, the value of the local optima within the attraction basins A is always 140.

3.1.2 Neighbourhood Relations. Moreover, the neighbourhood relation between solutions is preserved when the environment is rotated, no matter the degree of the rotation. Let's consider the neighbouring solutions $\sigma = 2130$ and $\sigma' = 2031$, $\sigma' \in N(\sigma)$ under the 2-exchange neighbourhood, as a practical case. Note that the neighbourhood of a solution does not necessarily belong to the same basins of attraction. After rotating the landscape by $\Pi = 0132$ (Figure 1(b)), the mapping of the solutions change to $f'(\sigma = 2130) = f_s(0132 \circ 2130) = f_s(3120)$ and $f'(\sigma' = 2031) = f_s(3021)$, respectively. Hence, their neighbourhood relation remains after rotating the landscape since $3120 \in N(3021)$.

3.1.3 Dynamics of the Landscape Rotation. Despite all that, although several works used the landscape rotation on the permutation domain [1, 16, 25], none studied the consequence of the generated dynamics. In this work, we empirically demonstrate that

Algorithm 1 sHC-R1

```

1: Let  $\sigma$  be a random permutation and  $e$  the identity permutation.
2:  $best \leftarrow$  Best solution found by  $sHC(e, \sigma)$ .
3: repeat
4:    $\Pi \leftarrow$  Uniformly at random permutation at distance  $d$ .
5:    $\sigma^* \leftarrow$  Best solution found by  $sHC(\Pi, best)$ .
6:   Update  $best$  if  $\sigma^*$  improves it.
7: until Stopping criterion is met.

```

the steepness of the landscape rotation goes beyond the rotation degree, i.e. even a small variation may have a large impact on the restructuring of the solution space. It can be deduced by the number of interchanged solutions between basins of attractions once the environment is modified, since it represents the probability of the algorithm exploring different areas when the landscape is rotated.

Let's illustrate the previous statement with the previous example. For the sake of easing the understanding, we highlight the migrated solutions respective to Figure 1(a) in Figures 1(b) and 1(c) in light blue. The landscape rotation in Figure 1(b) represents the interchange between the variables 2 and 3 in the problem instance. All solutions are rearranged to different positions compared to Figure 1(a), although only four solutions migrate to different basins: the solutions 2130 and 1230 move to basin B, while 2031 and 0231 move to basin A. However, rotating the original landscape by $\Pi = 1023$ produces the maximum inter-basin exchanges (as can be seen in Figure 1(c)), even though the landscape is also rotated by the swap between two variables. Hence, although both rotations are considered minor modifications of the original instance, their transcendence is completely different.

4 LANDSCAPE ROTATION AS A PERTURBATION STRATEGY

This section presents two rotation based perturbation strategies applied into the stochastic hill-climbing heuristic (sHC). These methods are designed to redirect the search of the algorithm by applying a set of variation operations to redistribute the fitness landscape. The following paragraphs are dedicated to exclusively describing perturbation strategies based on landscape rotation.

On the one hand, we design a profoundness rotation method (sHC-R1) analogous to the iterative local search (ILS) [13] or the variable neighbourhood search (VNS) [17]. Algorithm 1 shows the pseudo-code of this method for any minimisation problem. The algorithmic details of this strategy are described in the following paragraph.

sHC-R1 starts the search from a random solution σ , and continues to improve its quality by sHC, $sHC(e, \sigma)$, until a local optima is reached, $best$. The function $sHC(e, \sigma)$ returns the local optima, $best$, obtained after applying sHC on the original environment¹, starting from σ . Once a local optima is found, the environmental shift variable Π is generated uniformly at random, and the sHC is re-initialised from the previously found local optima by $sHC(\Pi, best)$. The function $sHC(\Pi, best)$ starts from $best$ and applies the sHC on the rotated environment $f_s(\Pi \circ \pi)$, returning a local optima σ^* . Note that the solutions on the rotated environment are mapped to

¹Note that $\forall \pi \in \mathbb{S}_n, \pi \rightarrow f_s(e \circ \pi) = f_s(\pi)$.

Algorithm 2 sHC-R2

```

1: Let  $\sigma$  be a random permutation and  $e$  the identity permutation.
2:  $best \leftarrow$  Best solution found by  $sHC(e, \sigma)$ .
3: repeat
4:    $\Pi \leftarrow$  Uniformly at random permutation at distance  $d$ .
5:    $\sigma^* \leftarrow$  Solution obtained by  $GreedySearch(\Pi, best, B_r)$ .
6:    $B_r \leftarrow B_r/2$ 
7:    $best \leftarrow$  Best solution found by  $sHC(\Pi, \sigma^*)$ .
8: until Stopping criterion is met.

```

a different objective value, $\forall \pi \in \mathbb{S}_n, \pi \rightarrow f_s(\Pi \circ \pi)$. The obtained local optima, σ^* , replaces the previously found $best$ solution if it improves its quality.

On the other hand, a broadness rotation procedure applied to a sHC (sHC-R2) is presented. The implementation of this technique is summarised in Algorithm 2. The strategy starts in the same manner as sHC-R1, although they differ on the manner they take advantage of the rotated environment. This method is designed to spend some time in the rotated environment before returning to the original instance and continuing to optimise the problem from a different location until a new local optima is found. The function $GreedySearch(\Pi, best, B_r)$ is initialised by $\Pi \circ best$, and continues the search on the rotated environments for a budget of iterations B_r . The 'time' to spend in the rotated instance, B_r , is halved as the search progresses. Then, the algorithm continues optimising by sHC until reaching a local optima. On the contrary to sHC-R1, the obtained local optima solution will be the starting point on the next iteration even if it is not the best overall solution found.

Our perception is that both methods are suitable to escape from plateaus or local optima and change the search direction of the algorithm. Moreover, we can deduce the behaviour of both strategies. sHC-R1 jumps between attraction basins going deeply until reaching a local optima, so it should stand out on instances with a small number of attraction basins. In contrast, sHC-R2 is more focused on the exploration of the solution space since it spends some time on the landscape rotation moving the search away from the initial point. That said, it may be a better option when the space is composed by a large number of attraction basins.

Apart from the algorithmic design of the methods, the degree in which the landscape is rotated also influences the search process of the algorithms. Both methods use the distance between permutations to control the shifting disturbance of the perturbation. Nevertheless, the rotation degree must be adjusted in accordance with the purpose of the algorithm at each moment (see Section 5.1). Some evolutionary algorithms use a cooling schedule to guide the algorithm from exploratory behaviour to exploitative behaviour [2, 9]. This balancing idea might be used in the landscape rotation by moving far away from a local optima at the beginning (high disturbance), and decreasing the distance until ending up moving towards a nearby location. Hence, the rotation is scaled and gradually decreased within the maximum distance and the minimum distance for a given metric. Mathematically, the linear cooling in terms of the rotation distance may be seen as

$$d = \left\lfloor \frac{i}{B} (d_{max} - d_{min}) + d_{max} \right\rfloor, \quad (5)$$

where i is the search iteration, B is the total budget of iterations, and d_{min} and d_{max} are the minimum and maximum distances of a metric, respectively.

Aforementioned, even a small perturbation in the local optima may lead to a different optimal solution. In other words, the neighbours of an optimal solution may belong to different attraction basins. Having said that, another strategy is to start with a high entropy and exponentially decrease the distance by

$$d = \left| d_{max} e^{\lambda i} \right|, \quad (6)$$

$$\lambda = \frac{\ln\left(\frac{d_{min}}{d_{max}}\right)}{B}, \quad (7)$$

where λ is the cooling rate.

5 EXPERIMENTATION

In this section, we describe the experiments carried out to study the profits of the rotation based algorithms on the QAP. First, we present the study performed to adjust the rotation distance and the neighbourhood-exploration budget (improvement trials) parameters for the rotation based algorithms. Second, we show and compare the performance of the previously tuned rotation based algorithms against other sHC versions.

5.1 Rotation Based Algorithm Calibration

It is worth noting that, in the case of the QAP, the landscape rotation only affects the distance matrix (see Equation 1). Furthermore, the minimal operation to change the objective value of a solution is the swap between two locations, whereas the maximum is a complete reorganisation of the variables.

For the sake of tuning the parameters of the rotation based algorithms up, we consider the following experimental setting:

- The QAP instances tai40a, tai40b and bur26a² [19], of size $n = 40$ and $n = 26$, respectively, are studied under the 2-exchange neighbourhood.
- The permutation distance metrics used in this study are Cayley, Hamming and Kendall's τ .
- We consider the linear and exponential cooling strategies (previously explained in Equations 5 & 6) along with constant rotation degrees.
- The number of neighbours to be compared before rotating, named *improvement trials*, is set to $n, n + 74, n + 148 \dots n + 740$. The chosen sequence represents 11 equally distributed points within the range between the problem size n and the exploration of the entire neighbourhood $780 = \binom{n}{2}$. Simultaneously, this variable also represents the "budget of time to spend on the rotated environment" variable, B_r , for sHC-R2.
- The stopping criterion is set to $B = 10^3 n$ iterations, and 30 independent runs are performed for each algorithm.

Figure 2 shows a detailed example of the behaviour of a single run of the rotation based algorithms with the exponential cooling under Cayley metric on tai40a. The horizontal axis represents the iteration of the search simultaneously on each plot. Furthermore, the three plots are organised in the following way. The upper chart

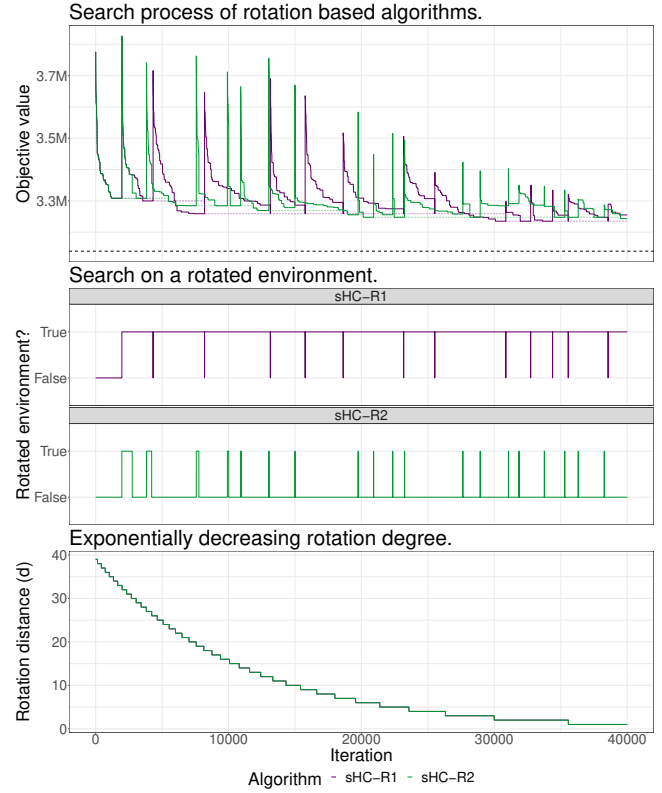


Figure 2: Information about the search of the rotation based algorithms in an independent run on tai40a.

shows the evolution of the algorithms, where the solid-coloured line represents the objective value of the candidate solution for each algorithm, the dotted-coloured line retains the best-found solution for each algorithm and the thick dashed line represents the best-known objective value. The middle plot shows whether the algorithm is searching in the original or on a rotated landscape in each iteration. Finally, the bottom part of the figure shows the distance at which the landscape may be rotated, measured by Cayley distance.

The top and bottom charts reflect the relationship between the rotation distance and the steepness of the modification. In general terms, the statement "a higher rotation distance means a larger perturbation" holds, since the environment is rotated with a higher entropy than it does at the end of the search. However, the third peak for sHC-R2 on the top graph goes further than the previous, even when the rotation distance is smaller (similar to the example shown in Figure 1).

Because of the infeasibility to display the performance of every algorithm for different parameter and metric configurations, results have been summarised as heatmaps in Figure 3. The tables are organised by metrics, trials and rotation distances for each of the algorithms. The colours in the tables are used for guidance only, where the lighter background colour, the better performance of the algorithm, i.e. the algorithm ends closer difference to the best-known solution in terms of objective value.

²Instances and their best-known values are obtained from <https://www.opt.math.tugraz.at/qaplib/inst.html>.

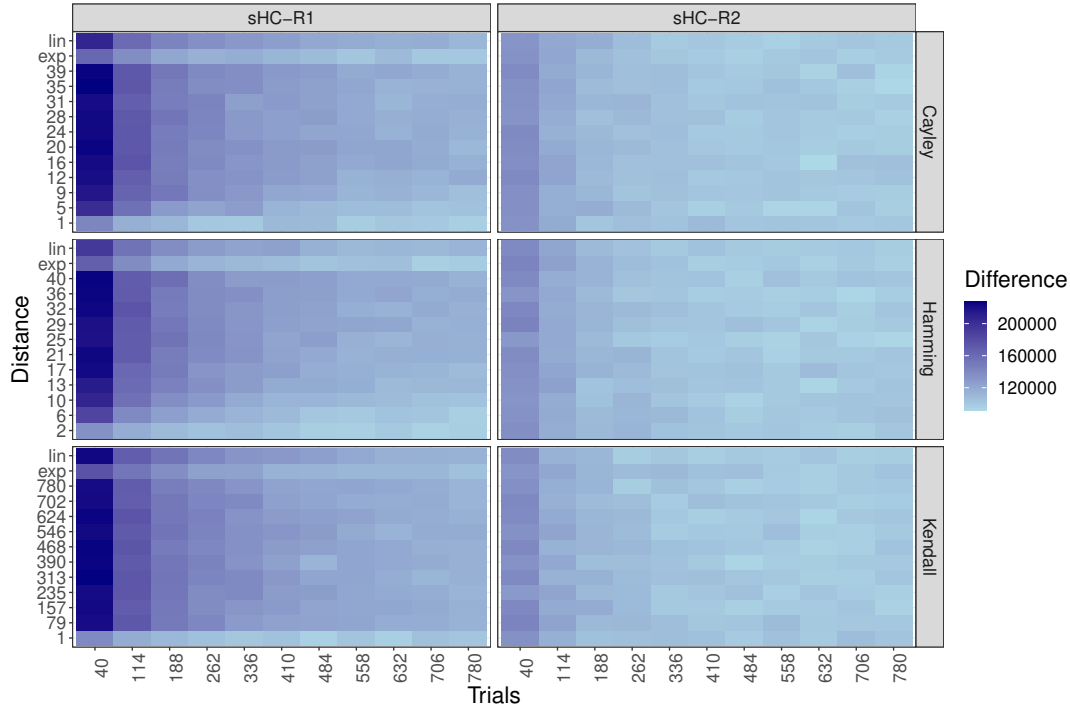


Figure 3: Difference between the mean performance of the algorithms to the best-known objective value for each configuration.

The performance of the algorithms does not change much from one metric to another. In other words, the colour gradient of the tables is similar for all the permutation metrics. That said, the performance of the algorithms differs depending on the number of improvement trials considered. The heatmaps for sHC-R1 reflect a darker tone on the left (small number of trials) of the heatmap than on the right (entire exploration of the neighbourhood). Therefore, it is worth exploring the entire neighbourhood before rotating the space, i.e. sHC-R1 should reach and compare local optima solutions for a good performance. Moreover, even rotating at the minimum distance for any metric is preferred to using cooling schemes. Nevertheless, the exponential cooling scheme is more efficient than the linear cooling to rotate the landscape. In sum, sHC-R1 seems to be a good method when exploring the entire neighbourhood, and applying slight rotations to the space. Alternatively, the tables for sHC-R2 display a similar colour gradient for all the configurations, although it reflects a better performance when spending some time exploring the neighbourhood before rotating the landscape. Thus, we can conclude that sHC-R2 benefits from the rotated environment no matter how and how much it is disturbed.

From the calibration, we conclude that there is no much difference between the types of modifications generated by the permutation metrics. In the following, we will see that the symmetry of the instances influences greatly the performance of the algorithms.

5.2 Experimental Study

For the sake of studying the applicability of the rotation based perturbations, we chose 11 instances from QAPLIB² with different properties and sizes [3, 12, 19, 26]. The parameter setting employed for the following experiments is described in Table 1.

In light of comparing the previously tuned algorithms, two other sHC variants are considered: the classic sHC and the multi-starting sHC (sHC-r). In case of the sHC-r, the number of improvement trials is the same as sHC-R1 and sHC-R2. Regardless, the rotation based parameters are deflected because the search is randomly reinitialised.

Figure 4 collects the average difference between the best solution found for each algorithm and the best-known value on instances with different symmetry structures³. The sub-captions also contain information about the characterisation (symmetries) of the distance and flow matrices of the selected QAP instances. Note that sHC

³Certain results are selected to highlight interesting aspects of the outcomes. Complete results on https://github.com/joanalza/GECCO_2021.git

Table 1: Parameter values for the experimental study.

Parameter	Value
Rotation degrees	$\{d_{min}, \dots, \lceil d_{max} * 0.25 \rceil\}$, linear & exponential
Number of improvement trials	$n, 2n, \dots, d_{max}^C$
Number of independent runs	30
Stopping criterion	$10^3 n$ iterations

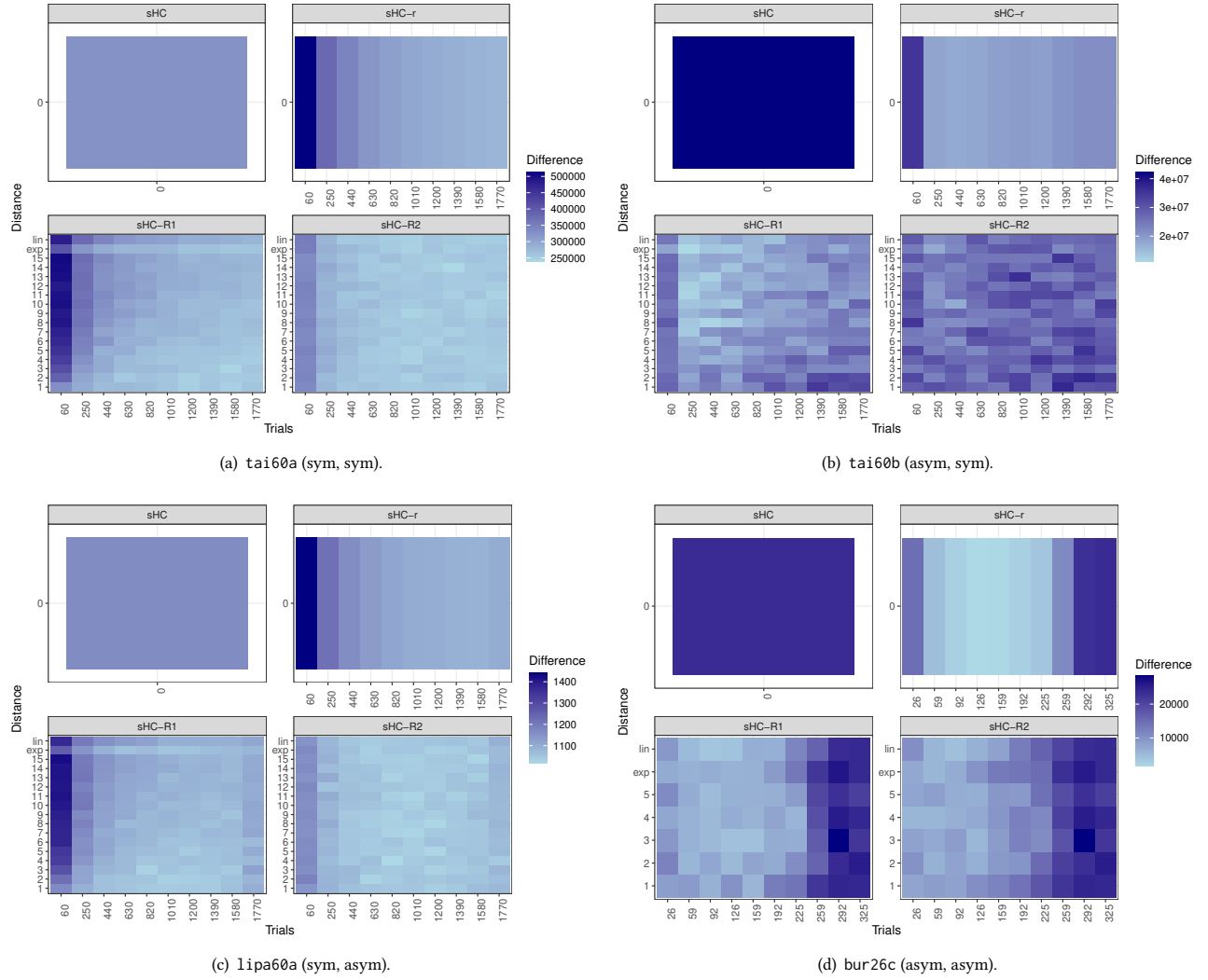


Figure 4: The performance of the algorithms sHC, sHC-r, sHC-R1 and sHC-R2, on four QAP instances with different symmetry structures.

only contains a single value as it does not react when the search gets stuck. Thus, it represents the average performance over the runs.

The plots point out that the benefits of the rotation based strategies differ from one instance to another, although they are generally good methods to avoid getting trapped on an attraction basins. As can be observed, the characterisation of the problem fairly affects the performance of the algorithms, and more specifically, the representation of the distance matrix. Figures 4(a) & 4(c) show the performance of the algorithms on instances with a symmetrical distance matrix. The plots reflect that the rotation based algorithms overcome the other sHC variants when rotating at close distance and exploring the entire neighbourhood (as presented in Section 5.1). Specifically, sHC-R2 acts better than sHC-R1 when the rotation parameters are tuned up. However, when the distance matrix is

asymmetric, Figures 4(b) & 4(d), the obtained results turn out to be more chaotic and very different from what was previously said.

In the case of the instance tai60b, which is composed of an asymmetric distance matrix and a symmetric flow matrix, sHC-R1 and sHC-R2 work worse than sHC-r in general, although better than sHC. In any event, sHC-R1 stands out when the rotated parameters are correctly tuned. In contrast, sHC-R1 and sHC-R2 perform in similar manner in the bur26c instance, which is completely asymmetric, i.e. the flow and distance matrices of the instance are both asymmetrical. Nevertheless, a simple restart is often more valuable than rotating the landscape. Having said that, we can deduce that the efficiency of the landscape rotation is affected by the characterisation of the distance matrix of the QAP instances.

As a curiosity, the linear and exponential cooling schemes presented in Equations 5 & 6 focus on the exploration of the space

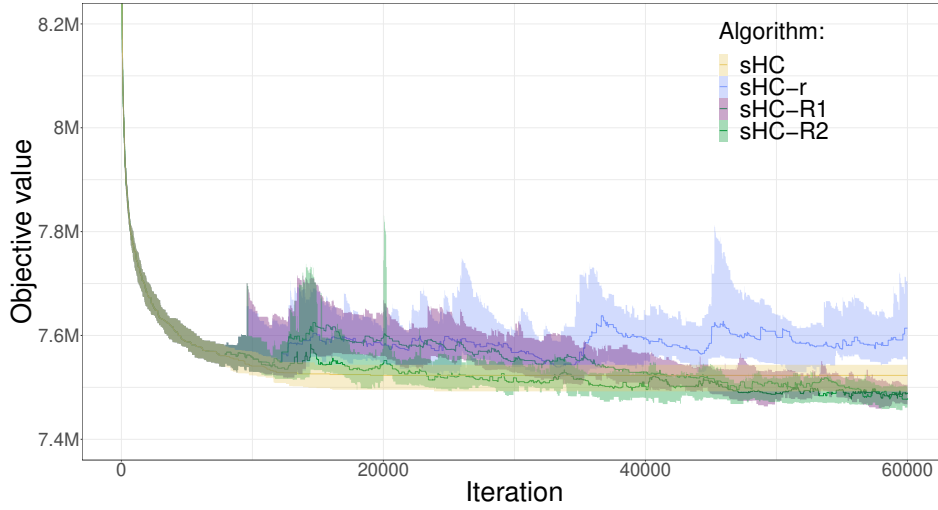


Figure 5: Interquartile range of the algorithms on tai60a under the exponential cooling and the exploration of the entire neighbourhood.

(initial iterations) in excess. Figure 5 shows the interquartile range of the search process of the algorithms over the runs on the tai60a instance under the entire exploration of the neighbourhood and the exponential cooling. The search of sHC gets trapped on the first quarter of the entire budget. Although the rest of the algorithms react to getting stuck, it is not until the middle (or even third quarter) of the search that the rotation based algorithms stand out against the sHC, that is, just when the environment rotates at a close distance. Therefore, results confirm that rotating the landscape constantly at close distances appears to be more profitable since a large rotation may result similar to a random restart due to the number of relabelled variables.

6 CONCLUSIONS AND FUTURE WORK

Landscape rotation was initially proposed in dynamic optimisation to generate a sequence of instances in order to test algorithms in a dynamic environment. This work shows that this idea may be also used as a perturbation technique in a local search heuristic to *kick* a search that gets trapped in an attraction basins of a local optima. Hence, this method changes the location of the solutions in the fitness landscape by mapping solutions to other objective values. Therefore, an already found local optima can be used as the starting point for the algorithm on the rotated environment in search of profitable areas of the space.

This work has proposed two perturbation strategies based on the landscape rotation to change the search direction of the algorithms in a controlled way. On the one hand, a profoundness rotation approach has been suggested. This strategy moves through local optima values by comparing them against new local optima found in rotated environments. On the other hand, a broadness rotation strategy was designed to kick a stuck search at some degree, and resume the search until reaching a new local optima.

Performed experiments have shown that rotating the landscape may be a good technique to perturb the space and search unexplored

areas of the solution space, although it needs to be cautiously manipulated. Generally speaking, the larger rotation distance means a bigger perturbation on the search process of an algorithm. However, the results have ratified that the rotation degree may be misleading as it does not necessarily modify the landscape in the wanted manner. This work has also demonstrated that even a small change may drastically change the fitness landscape in base of interchanges between basins of attraction. Therefore, we suggest using the landscape rotation in the permutation space in a cautious manner.

There are many interesting trends that may be extended from this work. One of the key things to study is the correspondence between basins of attraction and landscape rotation. Aforementioned, the consequence of the landscape rotation is related to the jumps of solutions between attraction basins after being rearranged. Thus, we think that analysing the number of attraction basins and their arrangement for instances with different characterisation may be a good contribution to the knowledge of landscape rotation in the permutation space. This work has made a contribution by applying two landscape rotation perturbation strategies on the stochastic hill-climbing heuristic to solve the QAP. This idea may be translated to other permutation problems such as the linear ordering problem (LOP) [4, 15], where the problem may be less sensitive to rotations. In the same direction, the permutation metric considered for the landscape rotation needs to be adjusted, since some metrics might be more suitable than others for each permutation problem. Finally, the landscape rotation may be applied to other metaheuristic algorithms to motivate the exploration of the space.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation (under the project PID2019-106453GA-I00/AEI/10.13039/501100011033), and the Consolidated Research Groups fund (IT1244-19) and the Elkartek Program (Project Code KK-2020/00049) from the Basque Government.

REFERENCES

- [1] Joan Alza, Mark Bartlett, Josu Ceberio, and John McCall. 2019. On the Definition of Dynamic Permutation Problems under Landscape Rotation. In *Proceedings of the GECCO* (Prague, Czech Republic). ACM, New York, NY, USA, 1518–1526.
- [2] Mayowa Ayodele, John McCall, Olivier Regnier-Coudert, and Liam Bowie. 2017. A Random Key based Estimation of Distribution Algorithm for the Permutation Flowshop Scheduling Problem. In *Proceedings in GECCO*. Association for Computing Machinery, New York, USA, 2364–2371.
- [3] R. Burkard and J. Offermann. 1977. Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme. *Zeitschrift für Operations Research* 21 (1977), B121–B132.
- [4] Josu Ceberio, Alexander Mendiburu, and Jose A. Lozano. 2015. The linear ordering problem revisited. *European Journal of Operational Research* 241, 3 (2015), 686–696.
- [5] Zvi Drezner. 2019. Taking advantage of symmetry in some quadratic assignment problems. *Information Systems and Operational Research* 57, 4 (2019), 623–641.
- [6] L. Hernando, A. Mendiburu, and J. A. Lozano. 2019. Anatomy of the Attraction Basins: Breaking with the Intuition. *Evolutionary Computation* 27 (2019), 435–466.
- [7] Ekhine Irurrozki. 2014. *Sampling and learning distance-based probability models for permutation spaces*. Ph.D. Dissertation. UPV/EHU.
- [8] R. Johnson and M. G. Pilcher. 1989. The traveling salesman problem, edited by E. L. Lawler, J. K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, John Wiley & sons, chichester, 1985, 463 pp. *Networks* 19, 5 (1989), 615–616.
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680.
- [10] Tjalling C. Koopmans and Martin Beckmann. 1957. Assignment Problems and the Location of Economic Activities. *Econometrica* 25, 1 (1957), 53–76.
- [11] Changhe Li, Shengxiang Yang, TT Nguyen, E Ling Yu, Xin Yao, Yaochu Jin, HG Beyer, and PN Suganthan. 2008. *Benchmark generator for CEC 2009 competition on dynamic optimization*. Technical Report. CEC competition.
- [12] Yong Li, Panos M. Pardalos, and Mauricio G.C. Resende. 1994. A Greedy Randomized Adaptive Search Procedure For The Quadratic Assignment Problem.
- [13] Helena R. Lourenço, Olivier C. Martin, and Thomas Stützle. 2003. *Iterated Local Search*. Springer US, Boston, MA, 320–353.
- [14] Katherine M. Malan and Andries P. Engelbrecht. 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241 (2013), 148–163.
- [15] Rafael Martí and Gerhard Reinelt. 2011. *The linear ordering problem: exact and heuristic methods in combinatorial optimization*. Vol. 175. Springer Science & Business Media, Berlin, Heidelberg.
- [16] Michalis Mavrovouniotis, Shengxiang Yang, and Xin Yao. 2012. A Benchmark Generator for Dynamic Permutation-Encoded Problems. In *PPSN*. Springer Berlin Heidelberg, Berlin, Heidelberg, 508–517.
- [17] N. Mladenović and P. Hansen. 1997. Variable neighborhood search. *Computers & Operations Research* 24, 11 (1997), 1097–1100.
- [18] Erik Pitzter and Michael Affenzeller. 2012. *A Comprehensive Survey on Fitness Landscape Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 161–191.
- [19] E. Taillard. 1991. Robust taboo search for the quadratic assignment problem. *Parallel Comput.* 17, 4 (1991), 443–455.
- [20] M. Tayarani-N. and A. Prügel-Bennett. 2014. On the Landscape of Combinatorial Optimization Problems. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 420–434.
- [21] Renato Tinós and Shengxiang Yang. 2010. An Analysis of the XOR Dynamic Problem Generator Based on the Dynamical System. In *Parallel Problem Solving from Nature, PPSN XI*. Springer Berlin Heidelberg, Berlin, Heidelberg, 274–283.
- [22] Renato Tinós and Shengxiang Yang. 2014. Analysis of fitness landscape modifications in evolutionary dynamic optimization. *Information Sciences* 282 (2014), 214–236.
- [23] Shengxiang Yang and Xin Yao. 2003. Dual population-based incremental learning for problem optimization in dynamic environments. *Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 49 – 56.
- [24] Shengxiang Yang and Xin Yao. 2005. Experimental Study on Population-Based Incremental Learning Algorithms for Dynamic Optimization Problems. *Soft Comput.* 9 (2005), 815–834.
- [25] Abdunnaser Younes, Paul Calamai, and Otman Basir. 2005. Generalized Benchmark Generation for Dynamic Combinatorial Problems. In *Proceedings in GECCO*. Association for Computing Machinery, New York, USA, 25–31.
- [26] Éric D. Taillard. 1995. Comparison of iterative searches for the quadratic assignment problem. *Location Science* 3, 2 (1995), 87–105.